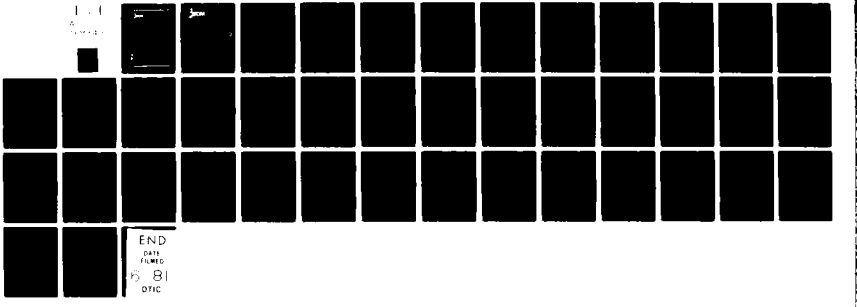
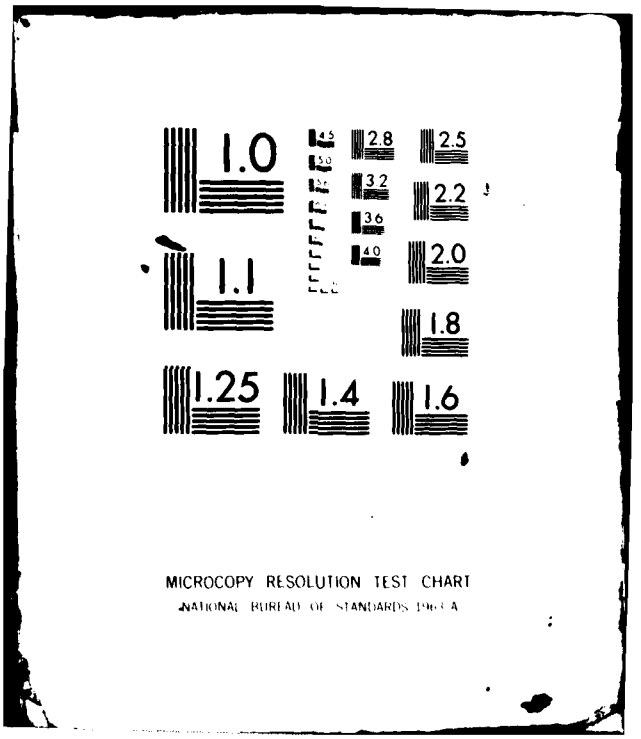
AD-A099 416 BDM CORP MONTEREY CA F/G 9/2
HP9825 PROGRAMS USED WITH THE PMS DAS-32 TO ACCUMULATE AND ANAL--ETC(U)
APR 81 D E SPIEL N00014-79-C-0088
UNCLASSIFIED BDM/M-006-81 NL



END
DATE FILMED
5 81
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

LEVEL II

12

THE BDM CORPORATION

P.O. BOX 2019 · 2600 GARDEN ROAD · MONTEREY, CALIFORNIA 93940 (408) 649-3880

AD A099416

FILE COPY

DTIC ELECTE MAY 28 1981 S D

A

81 5 01 05

THE BDM CORPORATION

P. O. Box 2019
2600 Garden Rd.
Monterey, California 93940
Phone (408) 649-3880

1 April 1981

Report No. BDM/M-006-81

HP9825 PROGRAMS USED WITH THE PMS DAS-32 TO ACCUMULATE AND ANALYZE AEROSOL DATA

D. E. Spiel
BDM/NPS
Monterey, California 93940

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: BDM/M-006-81 ✓	2. GOVT ACCESSION NO.: AD-A099 476
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): HP9825 Programs used with the PMS DAS-32 to Accumulate and Analyze Aerosol Data.	5. TYPE OF REPORT & PERIOD COVERED: Technical Report, Nov-Dec '80
	6. PERFORMING ORG. REPORT NUMBER: BDM/M-006-81
7. AUTHOR(s): D. E. Spiel, Project Engineer	8. CONTRACT OR GRANT NUMBER(s): N00014-79-C-0088 ✓
9. PERFORMING ORGANIZATION NAME AND ADDRESS: BDM Corporation - NPS Facility, 2600 Garden Road, Monterey, CA 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Work Order 315
11. CONTROLLING OFFICE NAME AND ADDRESS: Dean of Research, Code 012, Naval Postgraduate School, Monterey, CA 93940	12. REPORT DATE: 1 April 1981
	13. NUMBER OF PAGES: 37
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Mr. Robin Simpson, ONR Representative, Stanford University, Room 165, Durond Aeronautics Building, Stanford, CA 94305	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

See Initial Distribution List, page 37

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report was prepared under Work Order No. 315 of Contract No. N00014-79-C-0088 in support of a U.S. Naval Postgraduate School (NPS) research project sponsored by the Office of Naval Research, Code 460. The work was done in support of the Environemtnal Physics Group at NPS under the direction of Professors G. E. Schacher and K. L. Davidson.

DD FORM 1 JAN 73 1473 (Page 1) EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410 162

FOREWORD

This report was prepared under Work Order No. 315 of Contract No. N00014-C-0088 in support of a U.S. Naval Postgraduate School (NPS) research project sponsored by the Office of Naval Research, Code 460. The work was done in support of the Environmental Physics Group at NPS under the direction of Professors G. E. Schacher and K. L. Davidson.

TABLE OF CONTENTS

ABSTRACT

HP9825 computer programs for the acquisition and analysis of aerosol data are listed. The programs were written to be used with a PMS system in wave tank experiments, but can be used in other applications.

A. INTRODUCTION

This report briefly describes, defines the matrices used in, and lists a series of programs for the HP9825 used to accumulate and analyze aerosol data taken in conjunction with Particle Measuring Systems, Inc., (PMS) aerosol instruments. The programs were written specifically for use in wave tank experiments, but have a wider application.

The PMS instruments consist of a model DAS-32 data acquisition system and models ASASP and CSASP aerosol probes which, between them, scan the aerosol spectrum from 0.1 to 15 micrometers radius.

In addition to a data acquisition program, programs which integrate dN/dr spectra over various radius intervals are given. The remainder of the analysis programs are used to examine differences in spectra taken at different times and to look at these differences from several perspectives.

The programs described are:

(1) Data acquisition
(2) Plots log (dN/dr) vs. log(r)
(3) Plots Δ(log(dN/dr) vs. log (r)
(4) Tabulates Δ(log(dN/dr)) vs. r
(5) Plots log (Δ(dN/dr)) vs. log(r)
(6) Tabulates log(Δ(dN/dr)) vs. r
(7) Integrates dN/dr over r from 0.2, 0.3, 0.4, 0.6, 0.8 and 1 to 7 microns.
(8) Integrates dN/dr over r from 1.5, 2, 2.5, 3, 3.5 and 4 to 7 microns.
(9) Plots $(1/dN_1)(dN_2 - dN_1)$, the fractional change in dN/dr, vs. log(r).
(10) Plots the fractional change in dN/dr normalized byits peak value.
(11) Tabulates the aerosol density decay constant for various r.

B. PROGRAMS

1. Data Acquisition Program

This program provides the interfacing between the HP9825 computer and the PMS system. It accumulates data for a time period specified by the user and outputs it to a printer and a cassette tape. The data output to the printer is in both tabulated and plotted form. The tabulated data consists of raw counts vs. radius, dV/dr vs. radius and, at the option of the user, a table of the radius bin centers. The program calculates a polynomial fit, whose order is user specified (up to 10), of log(dV/dr) vs. log(r). This fit is plotted along with the individual data points (with some exclusions) by the HP9871A printer. A sample output is shown in Figure 1.

List of matrices:

Name	Function
A[*]	Accumulates sums used to calculate the polynomial fit.
B[*]	Denotes which printer symbols to be used for the six (6) radius ranges in the plot, except B[7] and B[8] used to extrapolate the fit.
C[*]	Cassette tape file bookkeeping, see below.
D	Current file number.
E[*]	Average radius within a bin. There are six ranges with 15 bins each.
F[*]	Accumulates sums used to calculate the polynomial fit.
G[*]	Coefficients of the polynomial fit.
H[*]	PMS analog housekeeping module outputs.
M[*]	Used in formulating extrapolations to polynomial curve.
N[*]	Radii range counter. Essentially keeps track of the total time the PMS was examining a given radii range.
O[*]	dN/dr and, later in the program, dV/dr.
P[*]	Sum of raw counts in each radius bin arranged in order of increasing radius.

R[*] Defines radii bin edges.

S[*] Sum of raw counts in each radius bin in the order specified by the DAS-32.

T[*] dN/dr

X[*] Defines which radius bins in each range are to be excluded from the polynomial fit calculations and the plot.

Y[*] Date and time, see below.

The components of C[*] are:

C[1] Number of files available on track 0 of the tape.

C[2] Number of files available on track 1 of the tape.

C[3] Tape number.

The components of Y[*] are:

Y[1] Year

Y[2] Month

Y[3] Day

Y[4] Hour

Y[5] Minute

Y[6] Second

Strings employed for outputing data, in split precision, to the tape are:

A$ Contains T[*], H[*], Y[*], X[*] and T, the averaging time.

Special function keys are employed as follows:

Key	Action	Purpose
0	* "cont formats"	Restarts programs without redefining matrices or loading keys
4	* sfg4	Branches program so that other special function keys may be set and acted upon.
5	* sfg5	Change order of polynomial.
6	* sfg6	Change averaging period.
7	* sfg7	Suppress all printing and plotting.
8	* cfg7	Restart printing and plotting.
9	* sfg9	Print radius bin center values
10	* sfg10	Mark a data tape.

Program Listing:

```
0: "
1: 
2: 
3: 
4: 
5: 
6: 
7: "
8: 
9: 
10: 
11: 
12: 
13: 
14: 
15: 
16: 
17: 
18: 
19: 
20: 
21: 
22: 
23: 
24: 
25: 
26: 
27: 
28: 
29: 
30: 
31: 
32: 
33: 
34: 
35: 
36: 
37: 
38: 
39: 
40: 
41: 
42: 
43: 
44: 
45: 
46: 
47: 
```

```
48: if J<3;int(X/100)+Y[2J-1];100frc(X/100)+Y[2J];if Y[1]=80;gto "data"
49: if J<5;gto "next"
50: X+W[J-4]+A[J-4];gto "next"
51: "bin":if flg1;B+1+B;jmp 3
52: if B<2;5+B;jmp 2
53: 6+B
54: X+S[B,I]+S[B,I]
55: "next":next I;1+N[B]+N[B];next J;cnf 2;1+F+F;if flg2;gto "loop"
56: if A<5;gto "data"
57: C[1]+C[2]+
58: if flg7;dsp "File",F,"Print successful.";jmp 2
59: dsp "Recording file",F,"max no.",
60: if [illegible];cll 'unmark'
61: for I=1 to 4;.[illegible][I]+[illegible][I];next I;30.4[illegible][5]+[illegible][5];30.4[illegible][6]+[illegible][6]
62: for I=1 to 5;for J=1 to 15;if S[I,J]=0;1+S[I,J]
63: next J;next I;for J=1 to 15;if S[6,J]=0;1+S[6,J]
64: next J
65: for J=1 to 15;S[1,J]/[illegible][1]+[illegible][4,J];S[2,J]/[illegible][2]+[illegible][3,J];S[3,J]/[illegible][3]+[illegible][2,J]
66: S[4,J]/[illegible][4]+[illegible][1,J];S[5,J]/[illegible][5]+[illegible][5,J];S[6,J]/[illegible][6]+[illegible][5,J]
67: S[1,J]+[illegible][4,J];S[2,J]+[illegible][3,J];S[3,J]+[illegible][2,J];S[4,J]+[illegible][1,J];S[5,J]+[illegible][5,J]
68: [illegible][5,J]+[illegible][5,J]
69: [illegible][J]/1000/[illegible]+[illegible][J];next J;[illegible][15]/1000/[illegible]+[illegible][15]
70: for I=1 to 5
71: for J=1 to 15;[illegible][I,J]/([illegible][I,J+1]-[illegible][I,J])+[illegible][I,J]+[illegible][I,J]
72: next J;next I
73: for I=1 to 3;for J=1 to 3;0+[illegible][I,J];next J;next I
74: [illegible][5,1]+[illegible][5,1];for J=1 to 15;[illegible][4,J];next J
75: [illegible]
76: [illegible]
77: "[illegible]":[illegible]("[illegible]"),255)+r3;[illegible](r3,15)+10[illegible](r3,4)+r2
78: [illegible]("[illegible]"),255)+r4;[illegible](r4,15)+10[illegible](r4,4)+10r2+r1;ret
79: "[illegible]":[illegible];for I=1 to 5;0+r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
80: for J=1 to 15;1+[illegible];if [illegible];1+r5
81: if J<15;[illegible]
82: if [illegible];gto "[illegible]"
83: if I=1;[illegible];gto "1"
84: if [illegible];log(r[illegible]/r[illegible])+[illegible];log(r7/r[illegible])+[illegible];1+r[illegible];gto "2"
85: 1.5+[illegible];gto "1"
86: if [illegible][I,J]=[illegible] r6=1;gto "next"
87: [illegible][I,J]^3*[illegible][I,J]+[illegible][I,J]
88: [illegible]r6=0;[illegible] "extra"
89: if [illegible][I,J][illegible] r6=1;gto "extra"
90: [illegible](S[I,J])+[illegible];[illegible]([illegible][I,J])+[illegible]
91: if I=1 and J<[illegible];gto "2"
92: if [illegible];gto "2"
93: gto "2"
94: "extra":r[illegible]+r9;r7+[illegible][I,J]+r7;r8+[illegible][I,J]+r8;gto "next"
95: "2":r0+X+r0;r1+[illegible]+r1;r2+1+r2;r3+X+r3;r4+[illegible]+r4;if r5=0;gto "2"
96: [illegible][7];[illegible][8];1.5+[illegible]
97: "1":(r0r2-r4r1)/(r0r2-r1r1)+[illegible][2]
98: (r[illegible]r[illegible]-r3r1)/(r0r2-r1r1)+[illegible][1];[illegible][1]+[illegible][2]*[illegible]
```

```
99: "[illegible]":for [illegible] to C;C+1-[illegible];[illegible][F,[illegible]]→[illegible][F,[illegible]]
100: [illegible][illegible]→[illegible];P*Y→[illegible];[illegible]
101: if [illegible];[illegible][[illegible]-1,[illegible]]→F[[illegible]-1,[illegible]]
102: [illegible];next [illegible]
103: "[illegible]":next [illegible]:next [illegible]
104: for I=1 to C+1
105: for K=1 to int((C+1)/2)
106: if [illegible] and I+K<C+2 and I-K>0;[illegible][I,[illegible]]→[illegible][I-K,I+K]→[illegible][I+K,I-K]
107: if I+K<=C+1 and I-K+1>0;[illegible][I,I+1]→[illegible][I+K,I-K+1]
108: if I+K<=C and I-K>0;[illegible][I,I+1]→[illegible][I-K,I+K+1]
109: next K;next I;inv [illegible];[illegible];if flg7;gto "[illegible]"
110: "out":wrt "[illegible].7",C[3],[illegible]
111: wrt "[illegible].3",Y[3],Y[3],Z[1],[illegible]Y[4]+Y[5],Z[5]
112: wrt "[illegible]",I;wrt "[illegible].5",R[1];wtb [illegible],10,13
113: wtb [illegible],"[illegible]/Cr"
114: wtb [illegible],10,13
115: for J=1 to 15
116: for I=2 to 5 by 3
117: wrt "[illegible].6",F[I,J]
118: next I;wtb [illegible],10,13;next J;wtb [illegible],10,13
119: if not flg9;jmp 3
120: wtb [illegible],"Radii";wtb [illegible],10,13;for J=1 to 15;for I=1 to 6;wrt "[illegible].6",E[I,J]
121: next I;wtb [illegible],10,13;next J;wtb [illegible],10,13;[illegible]
122: wtb [illegible],27,35,int(-3.75),int(-240),8,16,"[illegible](radius)"
123: wtb [illegible],27,[illegible],int(-1.[illegible]),int(-120),7,32,"[illegible](cm/Cr)"
124: "[illegible]":[illegible];[illegible]+1→[illegible];rot [illegible],[illegible],C[[illegible]];if [illegible]-1>C[1];[illegible] 1:[illegible]-C[1]-2→[illegible]
125: [illegible];if [illegible]7;gto "[illegible] print"
126: -1→[illegible];-[illegible]→[illegible]
127: wtb [illegible],27,[illegible],int(1[illegible]/[illegible]),int([illegible]),[illegible],[illegible])
128: if [illegible] or [illegible]=[illegible];wtb [illegible],"[illegible]",[illegible],[illegible],[illegible];wrt "[illegible].1",[illegible];gto +2
129: [illegible],"-"
130: if ([illegible]+[illegible])<[illegible];gto -3
131: wtb [illegible],27,[illegible],[illegible],int(1.5Y),int([illegible])
132: if [illegible] [illegible]=0;wrt "[illegible].2","-",[illegible];gto +2
133: wtb [illegible],"[illegible]"
134: if ([illegible]+[illegible])<[illegible].1;gto -2
135: for I=2 to [illegible] by 2
136: for J=1 to 15
137: if [illegible][I,J]=[illegible] or [illegible][I,J]=[illegible];gto "[illegible]"
138: [illegible]([illegible][I,J])→[illegible];[illegible]([illegible][I,J])→[illegible]
139: [illegible],27,[illegible],int([illegible]/[illegible]),int([illegible]),int([illegible]/[illegible]),int([illegible])
140: [illegible],[illegible]
141: "[illegible]":next [illegible];next [illegible];if [illegible];[illegible] 2
142: [illegible],27,[illegible],int([illegible][7]/[illegible]),int([illegible][7]),int(3[illegible][8]/2),int([illegible][8]),"[illegible]"
143: "[illegible]":for I=1 to [illegible];for [illegible]=1 to 15;[illegible]+15[illegible]-15→[illegible]
144: [illegible]([illegible][I,J])→[illegible][[illegible],[illegible]+4];next J;next I
145: [illegible]→[illegible];[illegible] I=[illegible] to [illegible]
146: [illegible]([illegible][I[illegible]])→[illegible][[illegible],3[illegible]+4K];1+[illegible]→[illegible];if I>[illegible];gto "[illegible]"
147: [illegible]([illegible][I])→[illegible][3[illegible],3[illegible]+4K];1+[illegible]→[illegible];if I>3;gto "[illegible]"
148: [illegible] ([illegible][[illegible]I-1])→[illegible][3[illegible]1+4K,362+4K]
149: [illegible] ([illegible][2I])→[illegible][3[illegible]3+4K,364+4K];1+K→K
```

```
150: "NEXT":next I
151: fts (I)→A$[457,460]
152: -1→X;0→Y;gto "gulp"
153: "crv":0→Y;if X>.8;gto "belch"
154: for I=1 to C+1;YX+C[I]→Y;next I
155: Y→M[4];jmp 2
156: "belch":if X>1.1;if M[2]>0;M[6]→Y;jmp 2
157: M[4]+M[2]*(X-.8)→Y→M[6]
158: if Y<-4 or Y>5;jmp 4
159: Y+6/96→Y
160: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
161: wtb M,"."
162: if (X+1/120→X)<1.5;gto "crv"
163: "gulp":wtb [illegible],12,13
164: [illegible]cf E,A$,[illegible];trk 0;rcf 0
165: "flags":if [illegible];gto "reset"
166: if [illegible] "poly order=",C," [illegible]";ent "",[illegible]
167: if [illegible] "[illegible] time=",[illegible],"[illegible]";ent "",[illegible];int([illegible])→[illegible]
168: [illegible] "[illegible] continue to begin";[illegible] gto "start"
169: "[illegible]":for I=1 to [illegible];wait 500;next I
170: [illegible] "Insert new cassette,continue.";[illegible]
171: [illegible] "[illegible] more?Continue.";[illegible]
172: [illegible] "[illegible]",[illegible]
173: [illegible]
174: [illegible];trk 1;[illegible];trk 0
175: if [illegible];gto "reset"
176: ret
*5909
```

2. Plot of log(dN/dr) vs. log(r)

This program retrieves the data stored on the cassette tape by the data acquisition program and produces a plot of log(dN/dr) vs. log(r). The plot consists of the individual data points, less those excluded, and a curve generated by a polynomial fit plus end point extrapolations. There is no tabulation of data.

The matrices used in this program serve the same function as in the data acquisition program.

The following is a listing of this program:

```
0: "BDM-32:PLOTS  LOG(dN/dr) FROM DATA IN FILES":
1: dim W$[64],A$[26]
2: dim S[5],X[5],P[5,15],T[15],C[5,15]
3: dim E[6],A[6,15],Z,Z[3]
4: dim P[6,15],A[10],P[10,10],J[10],A[6]
5: 35→S[1];111→S[2];43→S[3];0→S[4];42→S[5];64→S[6]
6: 715→I;dev "T",4
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,cl,f3.0,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0
11: fmt 4,/,"Averaging time = ",f3.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape #",f3.0," File #",f3.0," Event #"
14: fmt 8,"Polynomial of order ",f2.0,z
15: fmt 9,e16.7,z
16: 7→C
17: C+1→I;ron A[I],J[I],P[I,I]
18: dsp "Insert data tape,continue";stp
19: ldf 0,0,J[*]
20: ent "Track",I
21: dsp "printer on, set form,continue";stp
22: wtb 0,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
23: wtb 0,27,72,1,48,5,32
24: 0→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
25: .0345→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
26: .102→R[1,6];.1065→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
27: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
28: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
29: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
30: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
31: 1+(J-1)→R[6,J];.767*R[6,J]+.233→R[6,J];next J
32: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→E[I,J];next J;next I
```

```
33: "cycle":ent "Begin with file",A;ent "End with file",B
34: for D=A to B;fmt 0,10x,z;cll 'read tape'
35: for I=1 to 6
36: for J=1 to 15;T[I,J]→O[I,J];next J;next I
37: for I=1 to 5;for J=1 to 3;0→E[I,J];next J;next I
38: 0→O[5,1]→O[6,1];for J=1 to 15;0→O[4,J];next J
39: "rat":ino A,F;for I=1 to 5;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
40: for J=1 to 16;1→G;if I=6 and J#1;1→r6
41: if J<16;ino 5
42: if I>1 and I<6;gto "nEXt"
43: if I=1;-1.5→X;gto "1"
44: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
45: 1.5→X;gto "1"
46: if O[I,J]=0 and r6=0;gto "nEXt"
47: if r9#0;gto "extrap"
48: if O[I,J]=0 and r6=1;gto "extrap"
49: log(O[I,J])→P;log(E[I,J])→X
50: if I=1 and J<9;gto "3"
51: if I#6;gto "2"
52: gto "3"
53: "extrap":r9+1→r9;r7+O[I,J]→r7;r8+E[I,J]→r8;gto "nEXt"
54: "3":r0+X→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
55: X→P[7];P→P[6];1.5→X
56: "1":(r3r2-r4r1)/(r0r2-r1r1)→A[2]
57: (r0r4-r3r1)/(r0r2-r1r1)→A[1];A[1]+X*A[2]→P
58: "2":for K=0 to 6;6+1-K→N;G+P[P,N]→P[P,P]
59: P+A[P]→A[P];P*X→P;GG→G
60: if K<6;G+P[P-1,P]→P[P-1,P]
61: G*X→G;next K
62: "nEXt":next J;next I
63: for I=1 to 6+1
64: for K=1 to int((6+1)/2)
65: if I>1 and I+K<6+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
66: if I+K<=6+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
67: if I+K<=6 and I-K>0;F[I,I+1]→F[I-K,I+K+1]
68: next K;next I;inv F→F;mat FA→G;if flg7;gto "plt"
69: "out":wrt "M.7",G[3],D
70: wrt "M.3",Y[2],Y[3],Y[1],Y[4],Y[5]
71: wrt "M.4",P;wtb M,10,13
72: gto "no dN/dr"
73: wtb M,"dN/dr";wtb M,10,13;for J=1 to 15;for I=1 to 6
74: wrt "M.6",F[I,J];next I;wtb M,10,13;next J;wtb M,10,13
75: "no dN/dr":
76: wtb 1,27,65,int(-1.85),int(-120),7,32,"log(dN/dt)"
77: wtb 1,27,65,1,56,7,32;wrt "M.8",C;5.01-1/6→Y
78: for I=1 to 6+1;wtb 1,27,65,1,56,int(3Y/2),int(96Y)
79: wrt "M.9",3[I];Y-1/6→Y;next I
80: wtb M,27,65,int(-3.75),int(-240),0,16,"log(radius)"
```

```
81: "plt":if flg7;gto "skip print"
82: -1→X;-4→Y
83: wtb M,27,65,int(15X/4),int(240X),0,0
84: if X#0 and Xmod1=0;wtb M,"|",10,8,8,8;wrt "M.1",X;gto +2
85: wtb M,"-"
86: if (X+.05→X)<2.05;gto -3
87: wtb M,27,65,0,0,int(1.5Y),int(96Y)
88: if Y#0 and Ymod1=0;wrt "M.2","-",Y;gto +2
89: wtb M,"|"
90: if (Y+.1→Y)<5.1;gto -3
91: for I=1 to 6
92: for J=1 to 15
93: if D[I,J]=0 or O[I,J]=0;gto "NEXT"
94: log(D[I,J])→X;log(O[I,J])→Y
95: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
96: wtb M,S[I]
97: "NEXT":next J;next I;if r5=0;jmp 2
98: wtb M,27,65,int(152[7]/4),int(2403[7]),int(38[8]/2),int(963[3]),"e"
99: gto "skip print"
100: "read tape":trk M;ldf D,A$,E$
101: for I=1 to 6;for J=1 to 15;J+15I-16→K
102: stf(A$[4K+1,4K+4])→D[I,J];next J;next I
103: 0→K;for I=1 to 16
104: stf(A$[361+4K,364+4K])→X[I];1+K→K;if I>5;gto "NEXT"
105: stf(A$[361+4K,364+4K])→Y[I];1+K→K;if I>3;gto "NEXT"
106: itf(A$[361+4K,362+4K])→X[2I-1]
107: itf(A$[363+4K,364+4K])→X[2I];1+K→K
108: "NEXT":next I
109: stf(A$[457,460])→T
110: ret
111: "skip print":-1→X;0→Y
112: "crv":0→Y;if X>.3;gto "beloh"
113: for I=1 to C+1;YX+C[I]→Y;next I
114: Y→A[4];jmp 3
115: "beloh":if Y>1.1;if A[2]>0;A[6]→Y;jmp 2
116: A[4]+A[2]*(X-.3)→Y→A[6]
117: if Y<-4 or Y>5;jmp 4
118: Y+6/96→Y
119: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
120: wtb M,"."
121: if (X+1/120→X)<1.5;gto "crv"
122: wtb M,12,13
123: next D
124: gto "cycle"
125: end
*554
```

3. Plot Δ(log(dN/dr)) vs. log(r).

This program retrieves data from the cassette tape for two specified files, generates a polynomial fit to the data for each file, in log space, and plots the difference between the fits.

The matrices used in this program serve the same function as in the data acquisition program except that certain of them have been increased by one dimension in order to accommodate two data sets. K[*] are the coefficients of the polynomial fit to the second set of data.

A program listing follows:

```
0: "DMS-32: PLOTS DELTA LOG(dN/dr) FROM DATA ON FILE":
1: Dim A$[2,460],B$[460],P$[30],J[2],C[6,15],D$[25],Z[2]
2: Dim Y[2,6],X[6],Z[2,6,15],T[16],O[2,6,15]
3: Dim L[8],S[6,15],D,C[3],K[10]
4: Dim B[6,16],N[6,16],A[10],F[10,10],G[10],I[9],M[6]
5: 35→S[1];111→S[2];43→S[3];0→S[4];42→S[5];64→S[6]
6: 715→I;dev "4",4
7: Fmt 0,10x,z
8: Fmt 1,f3.0
9: Fmt 2,e1,f5.1,z
10: Fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 8,"Polynomial of order ",f2.0,z
15: Fmt 9,e15.7,z
16: 7→C
17: C+1→I;rdm A[I],G[I],R[I],F[I,I]
18: dsp "Insert data tape,continue";stp
19: ent "Tape #",C[3];ent "Track",J
20: dsp "Printer on, set form,continue";stp
21: wtb 3,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
22: wtb 3,27,79,4,48,6,32
23: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
24: .0845→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
25: .102→R[1,6];.1055→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
26: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
27: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
28: .2+.02(J-1)→R[3,J];.3+.03(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
29: if R[5,1]>1;.767*R[5,J]+.233→R[5,J]
30: 1+(J-1)→R[6,J];.767*R[6,J]+.233→R[6,J];next J
31: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→R[I,J];next J;next I
32: "start":ent "Event",C$;ent "Background file",J[1];ent "Event file",J[2]
```

```
33: trk 6;for D=1 to 2;lof J[D],BS,ES;BS$→A$[D];cll 'sort';next D
34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;T[D,I,J]→O[D,I,J];next J;next I
36: for I=1 to 3;for J=1 to 3;0→O[D,I,J];next J;next I
37: 0→O[D,5,1]→O[D,6,1];for J=1 to 15;0→O[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15;O[2,I,J]-O[1,I,J]→S[I,J]
39: if S[I,J]<0;0→S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ins A,F;for I=1 to 6;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
42: for J=1 to 16;1→S;if I=6 and J#1;1→r6
43: if J<15;jmp 5
44: if I>1 and I<6;gto "nExt"
45: if I=1;-1.5→X;gto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
47: 1.5→X;gto "1"
48: if O[D,I,J]=0 and r6=0;gto "nEXt"
49: if r9#0;gto "extrap"
50: if O[D,I,J]=0 and r6=1;gto "extrap"
51: log(O[D,I,J])→P;log(E[I,J])→X
52: if I=1 and J<9;gto "3"
53: if I#6;gto "2"
54: gto "3"
55: "extrap":r9+1→r9;r7+O[D,I,J]→r7;r8+E[I,J]→r8;gto "nEXt"
56: "3":r0+XX→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
57: X→B[7];P→B[8];1.5→X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)→M[2]
59: (r0r4-r3r1)/(r0r2-r1r1)→M[1];M[1]+X*M[2]→P
60: "2":for K=0 to C;C+1-K→R;G+F[R,R]→F[R,P]
61: P+A[R]→A[R];P*X→P;GX→G
62: if K#C;G+F[P-1,P]→F[P-1,R]
63: G*X→G;next K
64: "nEXt":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I#1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
68: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
69: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
70: next K;next I;inv F→F;if D=1;mat F→C;jmp 2
71: mat F→K
72: next D;ars K-G→S
73: "out":wrt "1.7",C[3],J[2],J[1],CS
74: Y[2,4]-Y[1,4]→Z;Y[2,5]-Y[1,5]+60*Z→Z
75: wrt "4.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: fmt 5," dT=",f3.0," minutes":wrt "1.5",Z
77: wrt "4.4",B[2],B[1];wtb 1,10,13
78: gto "no in/cr"
79: wtb 1,"dT/dr";wtb 1,10,13;for J=1 to 15;for I=1 to 6
```

```
80: wrt "M.5",T[I,J];next I;wtb M,10,13;next J;wtb M,10,13
81: "no dn/dr":
82: wtb M,27,65,-4,0,7,32,"delta[log(dN/dr)]"
83: wtb M,27,65,1,56,7,32;wrt "M.8",C;5.01-1/6→Y
84: for I=1 to C+1;wtb M,27,65,1,56,int(3X/2),int(96Y)
85: wrt "M.9",G[I];Y-1/6→Y;next I
86: wtb M,27,65,-4,-210,-1,-16,"log(radius)"
87: "plt":if flg7;gto "skip print"
88: -1→X;-4→Y
89: wtb M,27,65,int(15X/4),int(240X),0,0
90: if X#0 and Xmod1=0;wtb M,"|",10,8,3,3;wrt "M.1",X;gto +2
91: wtb M,"-"
92: if (X+.05→X)<2.05;gto -3
93: wtb M,27,65,0,0,int(1.5Y),int(96Y)
94: if Y#0 and Ymod1=0;wrt "M.2","-",.5*Y;gto +2
95: wtb M,"|"
96: if (Y+.1→Y)<5.1;gto -3
97: gto "skip print"
98: "sort":for I=1 to 6;for J=1 to 15;J+15I-16→K
99: str(A$[D,4K+1,4K+4])→T[D,I,J];next J;next I
100: 0→K;for I=1 to 16
101: stf(A$[D,361+4K,364+4K])→X[I];1+K→K;if I>5;gto "NEXT"
102: stf(A$[D,361+4K,364+4K])→Y[D,I];1+K→K;if I>3;gto "NEXT"
103: itf(A$[D,361+4K,362+4K])→X[2I-1]
104: itf(A$[D,363+4K,364+4K])→X[2I];1+K→K
105: "NEXT":next I
106: stf(A$[D,457,460])→Z[D]
107: ret
108: "skip print":-1→X;0→Y
109: "crv":0→Y
110: for I=1 to C+1;YX+G[I]→Y;next I
111: if Y<-4 or Y>5;jmp 4
112: 2*(Y+6/96)→Y
113: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
114: wtb M,"."
115: "help":if (X+1/128→X)<.85;gto "crv"
116: wtb M,12,13
117: gto "start"
118: end
*30256
```

4. Tabulation of Δ(log(dN/dr)) vs. r:

This program retrieves data from the cassette tape for two specified files and tabulates differences between them in two formats. In the first, a bin by bin difference between log(dN/dr) for the two files is tabulated vs. r. In the second, a tabulation of the differences between polynomial fits (as described in the preceding program) are given. The matrices serve the same functions as described above.

A program listing follows:

```
0: "DAS-32: ESTIMATES DELTA(LOG(dN/dr)) VS. r AND THE VALUES OF THE":
1: "ACCESS DATA FILES TO THIS FUNCTION FOR VARIOUS RADII":
2: dim A$[2,150],B$[460],C$[36],J[2],R[6,15],D$[25],Z[2]
3: dim Y[2,3],X[6],A[2,6,15],T[16],S[2,6,15]
4: dim P[2],N[6,15],O$[3],K[10],L$[25]
5: dim B[6,15],R[6,16],M[10],F[10,10],G[10],I[2],T[6]
6: "                              "→D$
7: 701→k;dev "T",1
8: fmt 0,10x,z
9: fmt 1,f3.0
10: fmt 2,c1,f4.0,z
11: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
12: fmt 6,e11.2,z
13: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 9,e15.7,z
15: 7→C
16: C+1→I;rdm A[I],G[I],K[I],R[I,1]
17: dso "Insert data tape,continue";stp
18: ent "Tape #",C[3];ent "Track",V
19: dso "Printer on, set form,continue";stp
20: wtb 6,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
21: wtb 6,27,79,4,43,6,32
22: .0845→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
23: .102→R[1,6];.1065→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
24: for J=11 to 15;R[1,J-1]+.005→R[1,J];next J
25: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
26: .24+.02(J-1)→R[3,J];.2+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
27: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
28: 1+(J-1)→R[6,J];.767*R[5,J]+.233→R[6,J];next J
29: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→X[I,J];next J;next I
30: "START":for V=1 to 2;wtb -1,10,10
31: ent "Event",C$;C$&D$[1,24-len(C$)]→C$
32: ent "Data file",T[1];ent "Event File",J[2]
```

```
33: trk 0;for D=1 to 2;ldf J[D],B$,C$;B$→A$[D];cll 'sort';next D
34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;F[D,I,J]→O[D,I,J];next J;next I
36: for I=1 to 3;for J=1 to 3;0→O[D,I,J];next J;next I
37: 0→O[D,5,1]→O[D,6,1];for J=1 to 15;0→O[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15;if O[2,I,J]=0 or O[1,I,J]=0;-1000→S[I,J];jmp 2
39: log(O[2,I,J])-log(O[1,I,J])→S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ina A,F;for I=1 to 6;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
42: for J=1 to 16;1→G;if I=6 and J#1;1→r6
43: if J<16;jmp 3
44: if I>1 and I<6;gto "nEXt"
45: if I=1;-1.5→X;gto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
48: if O[D,I,J]=0 and r6=0;gto "nEXt"
49: if r6#0;gto "extran"
50: if O[D,I,J]=0 and r6=1;gto "extra "
51: log(O[D,I,J])→P;log(S[I,J])→X
52: if I=1 and J<9;gto "3"
53: if I#6;gto "2"
54: gto "3"
55: "extran":r9+1→r9;r7+O[D,I,J]→r7;r8+F[I,J]→r8;gto "nEXt"
56: "3":r0+X→r0;r1+X→r1;r2+XX→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
57: X→F[7];P→F[8];1.5→X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)→B[2]
59: (r0r4-r3r1)/(r0r2-r1r1)→B[1];B[1]+X*B[2]→P
60: "2":for K=0 to C;C+1-K→R;G+F[R,R]→F[R,R]
61: P+B[R]→B[R];P*X→P;GX→G
62: if K#C;G+F[R-1,R]→F[R-1,R]
63: G*X→G;next K
64: "nEXt":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I>1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
68: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
69: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
70: next K;next I;inv F→E;if D=1;mat EB→G;jmp 2
71: mat EB→H
72: next D;mat H-G→G
73: "out":wrt "M.7",C[3],J[2],J[1],C$
74: Y[2,4]-Y[1,4]→Z;Y[2,5]-Y[1,5]+60*Z→Z
75: wrt "M.2",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: fmt 5,"  DT=",f3.0," minutes";wrt "M.5",Z
77: wtb 0,10,13
78: wtb 0,"delta(log(M/Mr))";wtb 0,10,13;for J=2 to 15;for I=1 to 6
79: if I=4;gto "OTDAY"
```

```
80: if S[I,J]<-999;wtb M,"      ",42,"      ";jmp 2
81: wrt "1.6",S[I,J]
82: "SAVE.AY":next I;wtb M,10,13;next J;wtb M,10,13
83: gto "CRV"
84: "sort":for I=1 to 6;for J=1 to 15;J+15I-15→K
85: ntf(AS[D,4K+1,4K+4])→Z[D,I,J];next J;next I
86: 0→K;for I=1 to 16
87: ntf(AS[D,361+4K,364+4K])→X[I];1+K→K;if I>5;gto "NeXT"
88: ntf(AS[D,361+4K,364+4K])→Y[D,I];1+K→K;if I>3;gto "NeXT"
89: ntf(AS[D,361+4K,362+4K])→X[2I-1]
90: ntf(AS[D,363+4K,364+4K])→X[2I];1+K→K
91: "NeXT":next I
92: ntf(AS[D,457,460])→Z[D]
93: ret
94: "CRV":for D=1 to 3;for J=1 to 5;.05(S(J-1)+D)-1.05→X;0→Y
95: for I=1 to C+1;YX+G[I]→Y;next I
96: Y→S[J,D]
97: next J;next D
98: wtb M,13,"Fitted Curve";wtb M,10,13;for J=1 to 6;for I=1 to 5
99: if S[I,J]=0;wtb M,"      ",42,"      ";jmp 2
100: wrt "1.6",S[I,J]
101: next I;wtb M,10,13;next J
102: next V
103: wtb M,13
104: gto "SIAFI"
105: end
*6233
```

5. Plot log (Δ(dN/dr)) vs. log(r)

This program retrieves data from two specified files on the cassette data tape and generates a polynomial fit, in log space, for each of the two files. It then plots, point by point, the logarithm of the difference between the antilogs of these two functions versus log(r). The matrices serve the same functions as described above.

A program listing follows:

```
0: "HAS-32: PLOTS LOG(DELTA(dN/dr)) FROM DATA ON FILE":
1: dim A$[2,460],B$[460],E$[36],J[2],C[6,15],C$[25],Z[2]
2: dim Y[2,6],X[6],P[2,6,15],T[15],D[2,6,15]
3: dim Z[2],E[6,15],D,C[3],K[10]
4: dim D[6,16],R[6,16],N[10],F[10,10],G[10],I[9],M[6]
5: 35→B[1];111→B[2];43→B[3];0→B[4];42→B[5];64→B[6]
6: 715→M;dev "T",4
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,c1,f4.0,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 8,"Polynomial of order ",f2.0,z
15: fmt 9,e15.7,z
16: 7→C
17: C+1→I;rdm A[I],G[I],K[I],F[I,I]
18: dsp "Insert data tape,continue";stp
19: ent "Tape #",C[3];ent "Track",J
20: dsp "Printer on, set form,continue";stp
21: wtb 1,27,69,27,34,32,32,32,32,27,77,27,76,13,0,14
22: wtb 1,27,79,4,48,6,32
23: 3→Y[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
24: .0845→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.096→R[1,5]
25: .102→R[1,6];.1035→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
26: for J=11 to 15;R[1,J-1]+.005→R[1,J];next J
27: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
28: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.23+.25(J-1)→P[5,J]
29: if P[5,J]>1;.767*P[5,J]+.233→R[5,J]
30: 1+(J-1)→P[6,J];.767*P[6,J]+.233→R[6,J];next J
31: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→E[I,J];next J;next I
32: "start":ent "Event",C$;ent "Background file",J[1];ent "Event file",J[2]
33: trk w;for D=1 to 2;ldf J[D],B$,E$;B$→A$[D];cll 'sort';next D
```

```
34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;T[D,I,J]→O[D,I,J];next J;next I
36: for I=1 to 3;for J=1 to 3;0→O[D,I,J];next J;next I
37: 0→O[D,5,1]→O[D,6,1];for J=1 to 15;0→O[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15;O[2,I,J]-O[1,I,J]→S[I,J]
39: if S[I,J]<0;0→S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ina A,F;for I=1 to 6;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
42: for J=1 to 15;1→G;if I=6 and J#1;1→r6
43: if J<15;jmp 5
44: if I>1 and I<6;gto "nEXt"
45: if I=1;-1.5→X;gto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
47: 1.5→X;gto "1"
48: if O[D,I,J]=0 and r6=0;gto "nEXt"
49: if r9#0;gto "extrap"
50: if O[D,I,J]=0 and r6=1;gto "extrap"
51: log(O[D,I,J])→P;log(F[I,J])→X
52: if I=1 and J<9;gto "3"
53: if I#6;gto "2"
54: gto "3"
55: "extrap":r9+1→r9;r7+O[D,I,J]→r7;r8+F[I,J]→r8;gto "nEXt"
56: "3":r0+XX→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
57: X→B[7];P→B[8];1.5→X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)→M[2]
59: (r0r4-r3r1)/(r0r2-r1r1)→M[1];M[1]+X*M[2]→P
60: "2":for K=0 to C;C+1-K→P;G+F[R,P]→F[P,P]
61: P+A[R]→A[P];P*X→P;GX→G
62: if K#C;G+F[R-1,P]→F[R-1,R]
63: G*X→G;next K
64: "nEXt":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I#1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
68: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
69: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
70: next K;next I;inv F→F;if D=1;mat FA→G;jmp 2
71: mat FA→K
72: next D
73: "out":wrt "%.7",C[3],J[2],J[1],CS
74: Y[2,4]-Y[1,4]→Z;Y[2,5]-Y[1,5]+60*Z→Z
75: wrt "%.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: fmt 5," JD=",f3.0," minutes";wrt "1.5",Z
77: wrt "%.4",Z[2],Z[1];wtb 4,10,13
78: gto "no dn/dt"
79: wtb M,"dX/dt";wtb 4,10,13;for J=1 to 15;for I=1 to 6
80: wrt "%.6",T[I,J];next I;wtb 4,10,13;next J;wtb M,10,13
81: "no dn/dt":
```

```
82: wtb M,27,65,-4,0,7,32,"log[delta(dY/dr)]"
83: wtb M,27,65,-4,-210,-1,-16,"log(radius)"
84: "plt":if flg7;gto "skip print"
85: -1→X;-4→Y
86: wtb M,27,65,int(15X/4),int(240X),0,0
87: if X#0 and Xmodl=0;wtb M,"|",10,8,8,8;wrt "M.1",X;gto +2
88: wtb M,"-"
89: if (X+.05→X)<2.05;gto -3
90: wtb M,27,65,0,0,int(1.5Y),int(96Y)
91: if Y#0 and Ymodl=0;wrt "M.2","-",Y;gto +2
92: wtb M,"|"
93: if (Y+.1→Y)<5.1;gto -3
94: gto "skip print"
95: "sort":for I=1 to 6;for J=1 to 15;J+15I-16→K
96: stf(A$[D,4K+1,4K+4])→H[D,I,J];next J;next I
97: 0→K;for I=1 to 16
98: stf(A$[D,361+4K,364+4K])→H[I];1+K→K;if I>5;gto "NeXt"
99: stf(A$[D,361+4K,364+4K])→Y[D,I];1+K→K;if I>3;gto "NeXT"
100: itf(A$[D,361+4K,362+4K])→X[2I-1]
101: itf(A$[D,363+4K,364+4K])→X[2I];1+K→K
102: "NeXT":next I
103: stf(A$[D,457,460])→Z[D]
104: ret
105: "skip print":-1→X
106: "crv":0→Y→Z
107: for I=1 to C+1;YX+G[I]→Y;ZX+K[I]→Z;next I
108: tn^2-tn^Y→Z;if Z<=0;gto "help"
109: log(Z)→Y
110: if Y<-4 or Y>5;jmp 4
111: Y+6/96→Y
112: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
113: wtb M,"."
114: "help":if (X+1/120→X)<.3;gto "crv"
115: wtb "",12,13
116: gto "start"
117: end
*24425
```

6. Tabulation of log (Δ(dN/dr)) vs. r

This program takes data from two specified files on the cassette tape and tabulates differences between them in two formats. In the first, the logarithm of the bin by bin differences between dN/dr is tabulated. In the second, a polynomial fit of log(dN/dr) vs. log(r) is generated for each of the two data sets and the logarithm of the differences of dN/dr, calculated from these two functions at specified radius values, is tabulated. The matrices employed are the same as in the previous programs.

A program listing follows:

```
0: "DMS-32: PROBLEMES LOG(DELTA(dN/dr)) vS. r AND THE VALUES OF":
1: "THE POLYNOMIAL FIT TO THIS FUNCTION FOR VARIOUS RADII":
2: dim $S[2,45],S$[45],S$[36],T[2],S[6,15],S$[25],Z[2]
3: dim X[2,6],X[5],T[2,5,15],R[15],S[2,5,15]
4: dim B[6],S[5,15],P,C[3],A[15],D$[25]
5: dim D[6,15],S[5,15],A[15],S[15,10],G[10],I[2],J[6]
6: "                              "→D$
7: 35→S[1];111→S[2];43→S[3];0→S[4];42→S[5];54→S[6]
8: 701→A;lev "?",?
9: fmt 0,10x,z
10: fmt 1,f3.0
11: fmt 2,c1,f4.0,z
12: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
13: fmt 6,e11.2,z
14: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
15: fmt 8,"Polynomial of order ",f2.0,z
16: fmt 9,e15.7,z
17: 7→C
18: C+1→I;rdm A[I],G[I],K[I],P[I,I]
19: dsp "Insert data tape,continue";stp
20: ent "Tape #",C[3];ent "Track",
21: dsp "Printer on, set form,continue";stp
22: wtb 6,27,69,27,84,32,32,32,32,27,77,27,75,15,0,14
23: wtb 7,27,79,4,43,6,32
24: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
25: .0045→P[1,1];.0075→P[1,2];.005→P[1,3];.004→P[1,4];.008→P[1,5]
26: .102→P[1,6];.1005→P[1,7];.111→P[1,8];.1155→P[1,9];.12→P[1,10]
27: for J=11 to 15;P[1,J-1]+.005→P[1,J];next J
28: for J=1 to 16;(.23+.025(J-1))/2→K[2,J]
29: .2+.02(J-1)→K[3,J];.34+.06(J-1)→K[4,J];.25+.25(J-1)→K[5,J]
30: if K[5,J]>1;.767*K[5,J]+.233→K[5,J]
31: 1+(J-1)→K[6,J];.767*K[6,J]+.233→K[5,J];next J
```

```
32: for I=1 to 6;for J=1 to 15;(P[I,J+1]+P[I,J])/2→S[I,J];next J;next I
33: "START":for V=1 to 2;wtb M,10,10
34: ent "Event",C$;C$&D$[1,24-len(C$)]→C$
35: ent "BACKGROUND FILE",J[1];ent "EVENT FILE",J[2]
36: trk W;for D=1 to 2;ldf J[D],B$,E$;B$→A$[D];cll 'sort';next D
37: for D=1 to 2;for I=1 to 6
38: for J=1 to 15;T[D,I,J]→O[D,I,J];next J;next I
39: for I=1 to 3;for J=1 to 3;0→O[D,I,J];next J;next I
40: 0→O[D,5,1]→O[D,6,1];for J=1 to 15;0→O[D,4,J];next J;next D
41: for I=1 to 6;for J=1 to 15;O[2,I,J]-O[1,I,J]→S[I,J]
42: if S[I,J]<=0;0→S[I,J];jmp 2
43: log(S[I,J])→S[I,J]
44: next J;next I;for D=1 to 2
45: "mat":ima A,F;for I=1 to 6;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
46: for J=1 to 16;1→G;if I=6 and J=1;1→r6
47: if J<16;jmp 5
48: if I>1 and I<6;gto "nEXt"
49: if I=1;-1.5→K;gto "1"
50: if I=6 and r6=0 and r7=0;log(r0/r1)→K;log(r7/r8)→Y;1→r5;gto "3"
51: 1.5→Y;gto "1"
52: if O[D,I,J]=0 and r6=0;gto "nEXt"
53: if r6=0;gto "extra"
54: if O[D,I,J]=0 and r6=1;gto "extra"
55: log(O[D,I,J])→X;log(S[1,J])→K
56: if I=1 and J<3;gto "3"
57: if I<6;gto "2"
58: gto "3"
59: "extra":r6+1→r6;r7+O[D,I,J]→r7;r8+P[I,J]→r8;gto "nEXt"
60: "3":r0+X→r0;r1+K→r1;r2+1→r2;r3+X→r3;r4+K→r4;if r5=0;gto "2"
61: X→S[7];K→S[8];1.5→K
62: "1":(r3r2-r4r1)/(r0r2-r1r1)→A[2]
63: (r0r4-r3r1)/(r0r2-r1r1)→A[1];S[1]+X*M[2]→P
64: "2":for K=0 to C;C+1-K→R;G+F[R,R]→F[R,R]
65: P+A[R]→A[P];P*X→P;GX→G
66: if K#C;G+F[R-1,R]→F[P-1,R]
67: G*X→G;next K
68: "nEXt":next J;next I
69: for I=1 to C+1
70: for K=1 to int((C+1)/2)
71: if I#1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
72: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
73: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
74: next K;next I;inv F→F;if D=1;mat FA→G;jmp 2
75: mat FA→K
76: next D
77: "out":wrt "V.7",C[3],J[2],J[1],C$
78: Y[2,4]-Y[1,4]→Z;Y[2,5]-Y[1,5]+60*Z→Z
79: wrt "N.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
```

```
80: fmt 5,"  dT=",f3.0," minutes";wrt "M.5",Z
81: wtb M,10,13
82: wtb M,"log[delta(dN/dr)]";wtb M,10,13;for J=2 to 15;for I=1 to 6
83: if I=4;gto "GALWAY"
84: if S[I,J]=0;wtb M,"      ",42,"      ";jmp 2
85: wrt "M.6",S[I,J]
86: "GALWAY":next I;wtb M,10,13;next J;wtb M,10,13
87: gto "CRV"
88: "sort":for I=1 to 6;for J=1 to 15;J+15I-15→K
89: str(A$[0,4K+1,4K+4])→P[0,I,J];next J;next I
90: 0→K;for I=1 to 16
91: str(A$[0,351+4K,364+4K])→H[I];1+K→K;if I>5;gto "NeXI"
92: str(A$[0,351+4K,364+4K])→Y[0,I];1+K→K;if I>3;gto "NeXI"
93: itr(A$[0,351+4K,352+4K])→X[2I-1]
94: itr(A$[0,353+4K,354+4K])→X[2I];1+K→K
95: "NeXI":next I
96: str(A$[0,457,460])→Z[1]
97: ret
98: "CRV":for J=1 to 3;for I=1 to 5;.05(3(J-1)+J)-1.05→X;0→Y→Z
99: for I=1 to 5+1;XK+3[I]→X;3K+K[I]→Z;next I
100: tn^Z-tn^Y→Z;if Z<=0;0→S[I,J];jmp 2
101: log(Z)→S[I,J]
102: next I;next J
103: wtb 1,13,"Fitted Curve";wtb M,10,13;for J=1 to 3;for I=1 to 5
104: if S[I,J]=0;wtb M,"      ",42,"      ";jmp 2
105: wrt "M.6",S[I,J]
106: next I;wtb M,10,13;next J
107: next V
108: wtb 1,13
109: gto "START"
110: end
*00120
```

7. Integration of dN/dr over r from 0.2 to 7 microns

This program integrates dN/dr over r from 0.2, 0.3, 0.4, 0.6, 0.8 and 1 to 7 microns for any specified file on the data tape. The integration employs the polynomial fit and the trapezoidal rule. The matrices serve the same function as in previous programs.

A program listing follows:

```
0: "INTEGRATES OVER dN/dr 0.2 TO 7 MICRONS GIVING TOTAL":
1: "N IN VARIOUS RADII INTERVALS. FROM DATA ON FILE":
2: dim A$[460],E$[36],U[6],3$[100]
3: dim Y[6],X[6],T[6,15],H[16],O[6,15]
4: dim B[8],E[6,15],D,C[3]
5: dim R[6,16],A[10],F[10,10],M[6],G[10]
6: 715→M;dev "M",1
7: 7→C
8: C+1→I:rdm A[I],G[I],F[I,I]
9: dsp "Insert data tape,continue";stp
10: ent "Tape #",C[3];ent "Track #",W
11: dsp "Printer on, set form,continue";stp
12: wtb M,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
13: wtb M,27,79,4,48,6,32
14: fmt 0,"TAPE #",f2.0," INTEGRATION OVER RADII."
15: wrt "M.0",C[3];fmt 0
16: wrt "M.0","SUM IS FROM INDICATED RADIUS TO 7 MICRONS.UNITS ARE CM^-3"
17: "  Date    Time    File    Sum(.2)    Sum(.3)    Sum(.4)    "→B$
18: B$&"Sum(.6)    Sum(.8)    Sum(1)"→B$;wtb M,B$,10,12
19: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
20: .0845→R[1,1];.0275→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
21: .102→R[1,6];.1065→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
22: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
23: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
24: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
25: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
26: 1+(J-1)→R[6,J];.767*R[6,J]+.233→R[6,J];next J
27: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→E[I,J];next J;next I
28: "start":ent "Begin with file",E;ent "End with file",F
29: for D=E to F;trk W;ldf D,A$,E$;cll 'sort'
30: for I=1 to 6
31: for J=1 to 15;T[I,J]→O[I,J];next J;next I
32: for I=1 to 3;for J=1 to 3;0→O[I,J];next J;next I
33: 0→O[5,1]→O[6,1];for J=1 to 15;0→O[4,J];next J
34: "mat":ima A,F;for I=1 to 6;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
35: for J=1 to 16;1→G;if I=6 and J#1;1→r6
```

```
36: if J<16;jmp 5
37: if I>1 and I<6;gto "nEXt"
38: if I=1;-1.5→X;gto "1"
39: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
40: 1.5→X;gto "1"
41: if O[I,J]=0 and r6=0;gto "nEXt"
42: if r9#0;gto "extrap"
43: if O[I,J]=0 and r6=1;gto "extrap"
44: log(O[I,J])→P;log(E[I,J])→X
45: if I=1 and J<9;gto "3"
46: if I#6;gto "2"
47: gto "3"
48: "extrap":r9+1→r9;r7+O[I,J]→r7;r8+E[I,J]→r8;gto "nEXt"
49: "3":r0+XX→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
50: X→B[7];P→B[8];1.5→X
51: "1":(r3r2-r4r1)/(r0r2-r1r1)→A[2]
52: (r0r4-r3r1)/(r0r2-r1r1)→A[1];A[1]+X*A[2]→P
53: "2":for K=0 to C;C+1-K→R;G+F[R,R]→F[R,R]
54: P+A[R]→A[R];P*X→P;GX→G
55: if K#C;G+F[R-1,R]→F[R-1,R]
56: G*X→G;next K
57: "nEXt":next J;next I
58: for I=1 to C+1
59: for K=1 to int((C+1)/2)
60: if I#1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
61: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
62: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
63: next K;next I;inv F→F;mat FA→G
64: gto "zisisit"
65: "sort":for I=1 to 6;for J=1 to 15;J+15I-16→K
66: stf(A$[4K+1,4K+4])→T[I,J];next J;next I
67: 0→K;for I=1 to 16
68: stf(A$[361+4K,364+4K])→X[I];1+K→K;if I>5;gto "JeXT"
69: stf(A$[361+K,364+4K])→X[I];1+K→K;if I>3;gto "JeXT"
70: itf(A$[361+4K,362+4K])→X[2I-1]
71: itf(A$[363+4K,364+4K])→X[2I];1+K→K
72: "JeXT":next I
73: stf(A$[457,460])→T
74: ret
75: "zisisit":
76: ina J;for P=.2 to 6.99 by .01;0→Y→Z;log(P)→A;log(P+.01)→E
77: for K=1 to C+1;Y+G[K]→Y;ZB+G[K]→Z;next K
78: tn^Y→Y;tn^Z→Z;.005*(Y+Z)→A;A+U[1]→U[1]
79: if P>.29;A+U[2]→U[2];if P>.39;A+U[3]→U[3];if P>.59;A+U[4]→U[4]
80: if P>.79;A+U[5]→U[5];if P>.99;A+U[6]→U[6]
81: next P;fmt 0,f2.0,"/",f2.0,"/",f2.0,f4.0,":",f2.0,z
82: fmt 1,f5.0,f11.2,f11.2,f11.2,f11.2,f11.2,f10.2
83: wrt "",Y[2],Y[3],Y[1],Y[4],Y[5]
84: wrt "",.1",0,U[1],U[2],U[3],U[4],U[5],U[6]
85: next D;gto "start"
86: end
*20070
```

8. Integration of dN/dr over r from 1.5 to 7 microns

This program is identical to the previous program except that it provides integrations of dN/dr over the intervals of r from 1.5, 2, 2.5, 3, 3.5 and 4 to 7 microns.

A program listing follows:

```
0: "DAS-32: INTEGRATES FROM 1.5,2,2.5,3,3.5,4 TO 7 MICRONS":
1: dim A$[460],C$[36],U[6],B$[100]
2: dim Y[6],X[6],T[6,15],N[16],O[6,15]
3: dim B[8],C[6,15],D,C[3]
4: dim P[6,16],A[10],F[10,10],M[6],G[10]
5: 715→K;dev "K",K
6: 7→C
7: C+1→I;rdm A[I],C[I],F[I,I]
8: dsp "Insert data tape,continue";stp
9: ent "Tape #",C[3];ent "Track #",J
10: dsp "Printer on, set form,continue";stp
11: wtb K,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
12: wtb K,27,79,4,48,6,32
13: fmt 0,"TAPE #",F2.0," INTEGRATION OVER RADII."
14: wrt "K.0",C[3];fmt 0
15: wrt "K.0","SUM IS FROM INDICATED RADIUS TO 7 MICRONS.UNITS ARE CM^-3"
16: " Date    Time    File    Sum(1.5)    Sum(2)    Sum(2.5)    "→B$
17: B$&"Sum(3)    Sum(3.5)    Sum(4)"→B$;wtb K,B$,10,13
18: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
19: .0345→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
20: .102→R[1,6];.1055→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
21: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
22: for J=1 to 15;(.23+.025(J-1))/2→R[2,J]
23: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
24: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
25: 1+(J-1)→R[6,J];.767*R[6,J]+.233→R[6,J];next J
26: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→E[I,J];next J;next I
27: "start":ent "Begin with file",B;ent "End with file",F
28: for D=B to F;trk K;lbf D,A$,D$;cll 'sort'
29: for I=1 to 6
30: for J=1 to 15;T[I,J]→O[I,J];next J;next I
31: for I=1 to 3;for J=1 to 3;0→O[I,J];next J;next I
32: 0→O[6,1]→O[6,1];for J=1 to 15;0→O[4,J];next J
33: "mat":ima A,F;for I=1 to 6;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
34: for J=1 to 15;1→G;if I=6 and J#1;1→r6
35: if J<16;jmp 5
36: if I>1 and I<6;gto "next"
37: if I=1;-1.5→X;gto "1"
```

```
38: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
39: 1.5→X;gto "1"
40: if O[I,J]=0 and r6=0;gto "nEXt"
41: if r9#0;gto "extrap"
42: if O[I,J]=0 and r6=1;gto "extrap"
43: log(O[I,J])→P;log(F[I,J])→X
44: if I=1 and J<9;gto "3"
45: if I#6;gto "2"
46: gto "3"
47: "extrap":r9+1→r9;r7+O[I,J]→r7;r8+O[I,J]→r8;gto "nEXt"
48: "3":r0+XX→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
49: X→B[7];P→B[8];1.5→X
50: "1":(r3r2-r4r1)/(r0r2-r1r1)→A[2]
51: (r0r4-r3r1)/(r0r2-r1r1)→M[1];M[1]+X*M[2]→P
52: "2":for K=0 to C;C+1-K→R;G+P[P,R]→F[R,R]
53: P+A[R]→A[R];P*X→P;GX→G
54: if K#C;G+P[R-1,R]→P[R-1,R]
55: G*X→G;next K
56: "nEXt":next J;next I
57: for I=1 to C+1
58: for K=1 to int((C+1)/2)
59: if I#1 and I+K<C+2 and I-K>0;P[I,I]→P[I-K,I+K]→P[I+K,I-K]
60: if I+K<=C+1 and I-K+1>0;P[I,I+1]→P[I+K,I-K+1]
61: if I+K<=C and I-K>0;P[I,I+1]→P[I-K,I+K+1]
62: next K;next I;inv P→P;mat PA→G
63: gto "zisisit"
64: "cort":for I=1 to 6;for J=1 to 15;J+15I-15→K
65: stf(A$[4K+1,4K+4])→P[I,J];next J;next I
66: 0→K;for I=1 to 16
67: stf(A$[361+4K,364+4K])→H[I];1+K→K;if D>5;gto "lexT"
68: stf(A$[361+4K,364+4K])→Y[I];1+K→K;if I>3;gto "lexT"
69: itf(A$[361+4K,362+4K])→X[2I-1]
70: itf(A$[363+4K,364+4K])→X[2I];1+K→K
71: "lexT":next I
72: stf(A$[457,460])→P
73: ret
74: "zisisit":
75: ina U;for P=1.5 to 6.38 by .02;0→Y→Z;log(P)→A;log(P+.02)→B
76: for K=1 to C+1;YA+G[K]→Y;ZB+G[K]→Z;next K
77: tn^Y→Y;tn^Z→Z;.01*(Y+Z)→A;A+U[1]→U[1]
78: if P>1.98;A+U[2]→U[2];if P>2.48;A+U[3]→U[3];if P>2.98;A+U[4]→U[4]
79: if P>3.48;A+U[5]→U[5];if P>3.98;A+U[6]→U[6]
80: next P;fmt 0,f2.0,"/",f2.0,"/",f2.0,f4.0,":",f2.0,z
81: fmt 1,f6.0,f12.4,f11.4,f11.4,f11.4,f11.4,f11.4
82: wrt "6",Y[2],Y[3],Y[1],Y[4],Y[5]
83: wrt "6.1",X,U[1],U[2],U[3],U[4],U[5],U[6]
84: next D;gto "start"
85: end
*14097
```

9. Plot Fractional Change in dN/dr

This program plots $(1/dN_1)(dN_2 - dN_1)$ as a function of log(r) from the data in any two specified files. It does so by utilizing polynomial fits, in log space, to the two data sets. The matrices employed have the same correspondence as in previous programs.

A program listing follows:

```
0: "DAS-32: PLOTS FRACTIONAL CHANGES IN dN/dr...(1/G)(DELTA(dN/dr))":
1: dim A$[2,460],B$[460],F$[36],J[2],S[6,15],C$[25],Z[2]
2: dim Y[2,6],X[6],I[2,6,15],B[15],O[2,6,15]
3: dim C[8],E[6,15],D,C[3],K[10]
4: dim D[6,15],R[6,16],A[10],F[10,10],G[10],I[9],M[6]
5: 35→B[1];111→B[2];43→B[3];0→B[4];42→B[5];64→B[6]
6: 715→M;dev "1",M
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,c1,f5.1,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 8,"Polynomial of order ",f2.0,z
15: fmt 9,e15.7,z
16: 7→C
17: C+1→I;dim A[I],G[I],K[I],F[I,I]
18: dsp "Insert data tape,continue";stp
19: ent "Tape #",C[3];ent "Track",M
20: dsp "Printer on, set form,continue";stp
21: wtb M,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
22: wtb M,27,79,4,48,6,32
23: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
24: .0845→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
25: .102→R[1,6];.1065→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
26: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
27: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
28: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
29: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
30: 1+(J-1)→R[6,J];.767*R[6,J]+.233→R[6,J];next J
31: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→E[I,J];next J;next I
32: "start":ent "Event",C$;ent "Background file",J[1];ent "Event file",J[2]
33: cre 7;for D=1 to 2;if J[D],S,C$;C$→A$[D];cll 'sort';next D
34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;I[D,I,J]→O[D,I,J];next J;next I
```

```
36: for I=1 to 3;for J=1 to 3;0→O[D,I,J];next J;next I
37: 0→O[D,5,1]→O[D,5,1];for J=1 to 15;0→O[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15;O[2,I,J]-O[1,I,J]→S[I,J]
39: if S[I,J]<0;0→S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ima A,F;for I=1 to 6;0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
42: for J=1 to 16;1→G;if I=6 and J#1;1→r6
43: if J<16;jmp 5
44: if I>1 and I<6;gto "nEXt"
45: if I=1;-1.5→X;gto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
47: 1.5→X;gto "1"
48: if O[D,I,J]=0 and r6=0;gto "nEXt"
49: if r9#0;gto "extrap"
50: if O[D,I,J]=0 and r6=1;gto "extrap"
51: log(O[D,I,J])→P;log(E[I,J])→X
52: if I=1 and J<3;gto "3"
53: if I#6;gto "2"
54: gto "3"
55: "extrap":r9+1→r9;r7+O[D,I,J]→r7;r8+E[I,J]→r8;gto "nEXt"
56: "3":r0+XX→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
57: X→S[7];P→S[8];1.5→X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)→A[2]
59: (r0r4-r3r1)/(r0r2-r1r1)→A[1];A[1]+X*A[2]→P
60: "2":for K=0 to C;C+1-K→R;G+S[R,R]→F[R,R]
61: P+A[R]→A[R];P*X→P;GX→G
62: if K#C;G+F[R-1,R]→F[R-1,R]
63: G*X→G;next K
64: "nEXt":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I#1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
68: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
69: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
70: next K;next I;inv F→F;if D=1;mat FA→G;jmp 2
71: mat FA→F
72: next D
73: "out":wrt "M.7",C[3],J[2],J[1],CS
74: Y[2,4]-Y[1,4]→Z;Y[2,5]-Y[1,5]+60*Z→Z
75: wrt "M.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: fmt 5,"  dT=",f3.0," minutes";wrt "M.5",Z
77: wrt "M.4",Z[2],Z[1];wtb M,10,13
78: wtb M,27,65,-4,0,7,32,"(1/N)(DELTA(dN/dR))"
79: wtb M,27,65,-4,-210,-1,-16,"log(radius)"
80: "plt":if flg7;gto "skip print"
```

```
81: -1→X;-4→Y
82: wtb M,27,65,int(15X/4),int(240X),0,0
83: if X#0 and Xmod1=0;wtb M,"|",10,8,8,8;wrt "M.1",X;gto +2
84: wtb M,"-"
85: if (X+.05→X)<2.05;gto -3
86: wtb M,27,65,0,0,int(1.5Y),int(96Y)
87: if Y#0 and Ymod1=0;wrt "M.2","-",2*Y;gto +2
88: wtb M,"|"
89: if (Y+.1→Y)<5.1;gto -3
90: gto "skip print"
91: "sort":for I=1 to 6;for J=1 to 15;J+15I-15→K
92: stf(A$[0,4K+1,4K+4])→T[0,I,J];next J;next I
93: 0→K;for I=1 to 15
94: stf(A$[0,361+4K,364+4K])→Y[I];1+K→K;if I>5;gto "NeXT"
95: stf(A$[0,361+4K,364+4K])→Y[0,I];1+K→K;if I>3;gto "NeXT"
96: itf(A$[0,361+4K,362+4K])→X[2I-1]
97: itf(A$[0,363+4K,364+4K])→X[2I];1+K→K
98: "NeXT":next I
99: stf(A$[0,457,460])→Z[0]
100: ret
101: "skip print":-1→X
102: "crv":0→Y;0→B
103: for I=1 to C+1;YX+G[I]→Y;3X+K[I]→Z;next I
104: tn^Z/tn^Y-1→T
105: if Y<-8 or Y>10;jmp 4
106: .5*(Y+6/96)→Y
107: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
108: wtb M,"."
109: "help":if (X+1/120→X)<.85;gto "crv"
110: wtb M,12,13
111: gto "start"
112: end
*17194
```

10. Plot Normalized Fractional Change in dN/dr

This program is identical to that just preceding except that the plot is normalized by the peak value of the fractional change in dN/dr. This is accomplished by determining the maximum value of the fractional change in dN/dr in the radius interval of interest and plotting the fractional change function divided by this maximum value. All such curves, then, have a maximum of one. The matrices are the same as in the previous program.

A program listing follows:

```
0: "DAS-32: PLOTS (1/N)(DELTA(dN/dr)) FOR ALLOWED BY ITS PEAK VALUE":
1: dim A$[2,460],B$[460],C$[36],J[2],G[6,15],C$[25],Z[2]
2: dim Y[2,6],X[6],P[2,6,15],R[15],D[2,6,15]
3: dim S[6],J[6,15],D,C[3],K[10]
4: dim O[6,16],P[6,16],A[10],F[10,10],G[10],I[9],W[6]
5: 35→O[1];111→S[2];43→S[3];0→J[4];42→B[5];64→S[6]
6: 715→L;dev " ",A
7: fmt 6,10x,z
8: fmt 1,f3.0
9: fmt 2,cl,f5.1,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 8,"Polynomial of order ",f2.0,z
15: fmt 9,e15.7,z
16: 7→C
17: C+1→I;rdm A[I],G[I],K[I],F[I,I]
18: dsp "Insert data tape,continue";stp
19: ent "Tape #",C[3];ent "Track",W
20: dsp "Printer on, set form,continue";stp
21: wtb M,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
22: wtb M,27,79,4,48,6,32
23: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
24: .0845→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
25: .102→R[1,6];.1055→R[1,7];.111→P[1,3];.1155→R[1,9];.12→R[1,10]
26: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
27: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
28: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
29: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
30: 1+(J-1)→P[6,J];.767*R[6,J]+.233→R[6,J];next J
31: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→D[I,J];next J;next I
32: "start":ent "Event",C$;ent "Background file",J[1];ent "Event file",J[2]
```

```
33: trk 0;for D=1 to 2;ldf J[D],B$,E$;B$+A$[D];cll 'sort';next D
34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;T[D,I,J]+O[D,I,J];next J;next I
36: for I=1 to 3;for J=1 to 3;0+O[D,I,J];next J;next I
37: 0+O[D,5,1]+O[D,6,1];for J=1 to 15;0+O[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15:O[2,I,J]-O[1,I,J]+S[I,J]
39: if S[I,J]<0;0+S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ina A,F;for I=1 to 6;0+r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
42: for J=1 to 15;1+G;if I=6 and J#1;1+r6
43: if J<15;jmp 5
44: if I>1 and I<6;gto "nEXt"
45: if I=1;-1.5+X;gto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)+X;log(r7/r9)+P;1+r5;gto "3"
47: 1.5+X;gto "1"
48: if O[D,I,J]=0 and r6=0;gto "nEXt"
49: if r9#0;gto "extrap"
50: if O[D,I,J]=0 and r6=1;gto "extrap"
51: log(O[D,I,J])+P;log(E[I,J])+X
52: if I=1 and J<9;gto "3"
53: if I#6;gto "2"
54: gto "3"
55: "extrap":r9+1+r9;r7+O[D,I,J]+r7;r8+E[I,J]+r8;gto "nEXt"
56: "3":r0+XX+r0;r1+X+r1;r2+1+r2;r3+PX+r3;r4+P+r4;if r5=0;gto "2"
57: X+B[7];P+B[8];1.5+X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)+M[2]
59: (r0r4-r3r1)/(r0r2-r1r1)+M[1];M[1]+X*M[2]+P
60: "2":for K=0 to C;C+1-K+R;G+F[R,R]+F[R,R]
61: P+A[R]+A[R];P*X+P;GX+G
62: if K#C;G+F[R-1,R]+F[R-1,R]
63: G*X+G;next K
64: "nEXt":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I#1 and I+K<C+2 and I-K>0;F[I,I]+F[I-K,I+K]+F[I+K,I-K]
68: if I+K<=C+1 and I-K+1>0;F[I,I+1]+F[I+K,I-K+1]
69: if I+K<=C and I-K>0;F[I,I+1]+F[I-K,I+K+1]
70: next K;next I;inv F+F;if D=1;mat FA+G;jmp 2
71: mat FA+K
72: next D
73: "out":wrt "M.7",C[3],J[2],J[1],C$
74: Y[2,4]-Y[1,4]+Z;Y[2,5]-Y[1,5]+60*Z+Z
75: wrt "M.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: fmt 5,"  DT=",f3.0," minutes";wrt "M.5",Z
77: wrt "M.1",Z[2],Z[1];wtb 7,10,13
78: wtb 7,27,65,-4,0,7,32,"<(1/Ro)(1/S)(DELTA(dV/dP))>"
79: wtb 7,27,65,-4,-210,-1,-16,"log(radius)"
```

```
80: "rlt":if flg7;gto "skip print"
81: -1→Y;-4→Y
82: wtb 1,27,65,int(15X/4),int(240X),0,0
83: if X#0 and Xtail=0;wtb 1,"|",10,8,3,4;wrt "^.1",X;gto +2
84: wtb 1,"-"
85: if (X+.05→X)<2.05;gto -3
86: wtb 1,27,65,0,0,int(1.5Y),int(96Y)
87: if Y#0 and Ytail=0; wrt "1.2","-",.2*Y;gto +2
88: wtb 1,"|"
89: if (Y+.1→Y)<5.1;gto -3
90: gto "skip print"
91: "sort":for I=1 to 6;for J=1 to 15;J+15I-16→K
92: stf(A$[D,4K+1,4K+4])→U[D,I,J];next J;next I
93: 0→K;for I=1 to 16
94: stf(A$[D,361+4K,364+4K])→W[I];1+K→K;if I>5;gto "NEXT"
95: stf(A$[D,361+4K,364+4K])→Y[D,I];1+K→K;if I>3;gto "NEXT"
96: itf(A$[D,361+4K,362+4K])→X[2I-1]
97: itf(A$[D,363+4K,364+4K])→X[2I];1+K→K
98: "NEXT":next I
99: stf(A$[D,457,460])→Z[D]
100: ret
101: "skip print":0→A;for B=1 to 2;-1→X
102: "crv":0→Y;0→%
103: for I=1 to C+1;YX+U[I]→Y;ZX+K[I]→Z;next I
104: tn^Z/tn^Y-1→Y;if B=2;jmp 3
105: if Y>A;Y→A
106: gto "help"
107: Y/A→Y;if Y<-1 or Y>1.1;jmp 4
108: 5*Y→Y
109: wtb 1,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
110: wtb 1,"."
111: "help":if (X+1/120→X)<.65;gto "crv"
112: next G
113: wtb 1,12,13
114: gto "start"
115: end
*3764
```

11. Tabulation of Aerosol Decay Constant

This program calculates the decay constant for aerosols as a function of radius. It assumes an exponential decay and requires the input from three data files, the third being the background. The time elapsed between the two post-event data sets must be specified. The calculations proceeds from polynomial fits in log space and is given by:

$$\tau = (t_2 - t_1)\,(\ln(dN_2/dr - dN_b/dr) - \ln(dN_1/dr - dN_b/dr))$$

Where t_i is the time of the $i^{\underline{th}}$ file and the subscript b denotes background.

The matrices employed are the same as in foregoing programs except the dimension of some arrays has been increased by one to accommodate inclusion of the background data set.

A program listing follows:

```
0: "DAS-32: TABULATES THE AEROSOL DECAY CONSTANT, TAU, FOR VARIOUS RADII":
1: dim A$[3,460],B$[460],E$[36],J[4],C$[25],Z[3];eig 14
2: dim Y[3,6],X[6],H[16],O[3,6,15]
3: dim D[6,15],D,C[3],K[10],S[6,15]
4: dim R[6,16],A[10],F[10,10],G[10],I[10],M[6],D$[25]
5: "                              "→D$
6: 701→M;dev "M",M
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,cl,f4.0,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape ",f2.0," Skand ",f3.0," Files ",f3.0," & ",f3.0," dT=",f5.2
14: fmt 9,e15.7,z
15: 7→C
16: C+1→I;ran M[I],G[I],X[I],I[I],F[I,I]
17: dsp "Insert data tape,continue";stp
18: ent "Tape #",C[3];ent "Track",N
19: dsp "Printer on, set form,continue";stp
20: wtb 1,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
21: wtb 1,27,79,4,43,6,32
22: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
```

```
23: .0845→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
24: .102→R[1,6];.1065→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
25: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
26: for J=1 to 15;(.23+.025(J-1))/2→R[2,J]
27: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
28: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
29: 1+(J-1)→R[6,J];.767*R[6,J]+.233→R[6,J];next J
30: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→E[I,J];next J;next I
31: "START":for V=1 to 2;wtb M,10,10
32: "start":ent "Event",C$;C$&DS[1,24-len(C$)]→C$
33: ent "BACKGROUND FILE?",J[1];if J[1]=J[4];stg 1
34: J[1]→J[4]
35: ent "BEGIN FILE",J[2];ent "ENDING FILE",J[3];ent "TAPE DIF?",P
36: trk M;for D=1 to 3;if flg1;cfg 1;jmp 2
37: Lif J[D],.3,DS;DS→AS[D];cll 'sort'
38: next D;for I=1 to 6;for J=1 to 15
39: O[2,I,J]-O[1,I,J]→O[2,I,J];O[3,I,J]-O[1,I,J]→O[3,I,J]
40: if O[3,I,J]=0;-10000→S[I,J];gto "BELFAST"
41: O[2,I,J]/O[3,I,J]→O[2,I,J]→S[I,J]
42: if S[I,J]<=0;-10000→S[I,J];jmp 2
43: T/ln(S[I,J])→S[I,J]
44: "BELFAST":next J;next I;for I=2 to 3;0→C[3,1,I]→O[3,2,I];next I
45: for I=2 to 3;-10000→S[1,I]→S[2,I];next I
46: for D=2 to 2
47: "mat":ina A,F;for I=1 to 6;if I=4;gto "nEXtI"
48: for J=2 to 15;1→G
49: if O[3,I,J]=0;gto "nEXtJ"
50: O[D,I,J]→P;log(E[I,J])→K
51: "2":for R=0 to C;C+1-K→R;C+F[P,R]→F[R,R]
52: P+A[P]→A[P];P*X→P;GX→G
53: if R<C;C+F[P-1,R]→F[P-1,R]
54: G*X→G;next K
55: "nEXtJ":next J
56: "nEXtI":next I
57: for I=1 to C+1
58: for K=1 to int((C+1)/2)
59: if I>1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
60: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
61: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
62: next K;next I;inv F→F;if D=1;mat F→G;jmp 3
63: if D=2;mat F→K;jmp 2
64: mat F→I
65: next D
66: "out":wrt "%.7",C[3],J[1],J[2],J[3],P
67: wrt "%.2",Y[1,2],Y[1,3],Y[1,1]
68: fmt "%",Event",c25;wrt "%",C$;wtb 11,10,13
69: wtb 1,"Tau (minutes)";wtb 11,10,13;for J=2 to 15;for I=1 to 6
```

```
70: if I=4;gto "GALWAY"
71: if S[I,J]<-9999;wtb M,"      ",42,"    ";jmp 3
72: if S[I,J]>9e99;wtb M,"      ",42,"    ";jmp 2
73: wrt "M.6",S[I,J]
74: "GALWAY":next I;wtb M,10,13;next J:wtb M,10,13
75: gto "crv"
76: wtb M,27,65,0,0,int(1.5Y),int(96Y)
77: if Y#0 and Ymod1=0;wrt "M.2","-",2Y;gto +2
78: wtb M,"|"
79: if (Y+.1→Y)<5.1;gto -3
80: gto "skip print"
81: "sort":for I=1 to 6;for J=1 to 15;J+15I-16→K
82: stf(A$[D,4K+1,4K+4])→S[D,I,J];next J;next I
83: 0→K;for I=1 to 16
84: stf(A$[D,361+4K,364+4K])→H[I];1+K→K;if I>5;gto "NeXT"
85: stf(A$[D,361+4K,364+4K])→Y[D,I];1+K→K;if I>3;gto "NeXT"
86: itf(A$[D,361+4K,362+4K])→X[2I-1]
87: itf(A$[D,363+4K,364+4K])→X[2I];1+K→K
88: "NeXT":next I
89: stf(A$[D,457,460])→Z[D]
90: ret
91: "skip print":-1→X
92: "crv":for D=1 to 8;for J=1 to 5;.05(3(J-1)+2)-1.05→X;0→Y
93: for I=1 to CH;YX+S[I]→Y;next I
94: Y→S[J,0]→Y;if Y<=0;-10000→S[J,0];jmp 2
95: Y/ln(2)→S[J,0]
96: next J;next D
97: wtb M,13,"Fitted Curve";wtb M,10,13:for D=1 to 8;for I=1 to 5
98: if S[I,J]<-9999;wtb M,"      ",42,"    ";jmp 2
99: wrt "M.6",S[I,J]
100: next I;wtb M,10,13;next J
101: next V
102: wtb M,13
103: gto "START"
104: end
*31767
```

DISTRIBUTION LIST

		DODAAD Code	No. of Copies
1.	Mr. Robin Simpson, ONR Rep. Stanford University, Room 165 Durond Aeronautics Building Stanford, California 94305		1
2.	Dean of Research, Code 012 Naval Postgraduate School Monterey, California 93940	N62271	3
3.	Defense Documentation Center Cameron Station, Building 5 Alexandria, Virgina 22314	S47031	12
4.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940		
5.	Director, Naval Research Lab. Attn: Code 2627 Washington, D.C. 20375	N00173	6
6.	Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91106	N62879	1
7.	Associate Professor K. Davidson Naval Postgraduate School Code 63Ds Monterey, California 93940		1
8.	Professor Gordon Schacher Naval Postgraduate School Code 61sq Monterey, California 93940		1


DAT
FILM